

1 **CLAIMS**

2 **1.** A method comprising:

3 establishing a secure communication channel between a media playback
4 application and a component downstream from the media playback application;
5 and

6 using the secure communication channel to at least enable the media
7 playback application to instruct the downstream component to enable one or more
8 of a number of different types of content protection technologies to protect media
9 content that is provided over a physical connector.

10
11 **2.** The method of claim 1 further comprising using the secure communication
12 channel to enable the media playback application to instruct the downstream
13 component as to how to apply one or more of the different types of content
14 protection technologies.

15
16 **3.** The method of claim 1, wherein the downstream component comprises a
17 software component.

18
19 **4.** The method of claim 1 further comprising using the secure communication
20 channel to enable the media playback application to request status information
21 from the downstream component.

1
2 **5.** The method of claim 1 further comprising:

3 using the secure communication channel to enable the media playback
4 application to request status information from the downstream component; and

5 using the secure communication channel to receive status information from
6 the downstream component.
7

8 **6.** The method of claim 1 further comprising:

9 using the secure communication channel to enable the media playback
10 application to request status information from the downstream component; and

11 using the secure communication channel to receive status information from
12 the downstream component, wherein the status information pertains to instructions
13 that were previously sent by the media playback application.
14

15 **7.** The method of claim 1 further comprising:

16 using the secure communication channel to enable the media playback
17 application to request status information from the downstream component; and

18 using the secure communication channel to receive status information from
19 the downstream component, wherein the status information does not pertain to
20 instructions that were previously sent by the media playback application.
21

22 **8.** One or more computer-readable media having computer-readable
23 instructions which, when executed, implement the method of claim 1.
24
25

1 **9.** A computing system embodying the one or more computer-readable media
2 of claim 8.

3
4 **10.** A system comprising:
5 one or more computer-readable media;
6 a software component resident on the media and configured to:
7 establish a secure communication channel with a media playback
8 application;
9 use the secure communication channel to receive instructions from
10 the media playback application to enable one or more of a number of
11 different types of content protection technologies to protect media content
12 that is provided over a physical connector; and
13 for at least some of the content protection technologies, receive
14 instructions to configure the content protection technologies.

15
16 **11.** The system of claim 10, wherein the software component comprises a
17 software driver.

18
19 **12.** The system of claim 10, wherein the software component is further
20 configured to use the secure communication channel to receive status requests
21 from the media playback application.
22
23
24
25

1 **13.** The system of claim 10, wherein the software component is further
2 configured to use the secure communication channel to receive status requests
3 from the media playback application, and wherein the software component is
4 further configured to use the secure communication channel to send status
5 information to the media playback application.

6
7 **14.** The system of claim 10, wherein the software component is further
8 configured to use the secure communication channel to receive status requests
9 from the media playback application, and wherein the software component is
10 further configured to use the secure communication channel to send status
11 information to the media playback application, wherein the status information
12 pertains to instructions that were previously received from the media playback
13 application.

14
15 **15.** The system of claim 10, wherein the software component is further
16 configured to use the secure communication channel to receive status requests
17 from the media playback application, and wherein the software component is
18 further configured to use the secure communication channel to send status
19 information to the media playback application, wherein the status information
20 does not pertain to instructions that were previously received from the media
21 playback application.

22
23 **16.** A computing system embodying the system of claim 10.
24
25

1 17. A method comprising:
2 establishing trust between a media playback application and a downstream
3 component;
4 establishing a secure channel between the media playback application and
5 the downstream component using a public key associated with the downstream
6 component to encrypt:
7 a random number provided by the downstream component;
8 a data integrity key; and
9 one or more starting numbers;
10 sending the encrypted data to the downstream component;
11 using the secure channel to send a command message to the downstream
12 component, the command message comprising a data section that contains a
13 command, and an authentication section that contains data that can be used to
14 authenticate the command;
15 using the secure channel to request status information from the downstream
16 component; and
17 using the secure channel to receive a status message from the downstream
18 component, the status message comprising a data section that contains status
19 information, and an authentication section that contains data that can be used to
20 authenticate the status information.
21
22
23
24
25

1 **18.** The method of claim 17, wherein said one or more starting numbers
2 comprise a starting status sequence number and a starting command sequence
3 number, said numbers being useable to ascertain, respectively, whether a status
4 message or a command message has been lost.

5
6 **19.** The method of claim 17, wherein the act of using the secure channel to
7 request status information from the downstream component comprises sending,
8 with the request, a random number, and wherein the authentication section of the
9 status message comprises data associated with the random number.

10
11 **20.** The method of claim 17, wherein the authentication sections of the
12 command message and the status message comprise data that has been processed
13 using the data integrity key.

14
15 **21.** The method of claim 17, wherein the command message contains a
16 command instructing the downstream component to enable one or more of a
17 number of different types of content protection technologies to protect media
18 content that is provided over a physical connector.

19
20 **22.** The method of claim 17, wherein the downstream component comprises a
21 software driver.

22
23 **23.** One or more computer-readable media having computer-readable
24 instructions which, when executed, implement the method of claim 17.
25

1 **24.** A computing system embodying the one or more computer-readable media
2 of claim 23.

3
4 **25.** The method of claim 17 further comprising using the secure channel to
5 provide protected media content to the downstream component.

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

1 **26.** A system comprising:
2 one or more computer-readable media;
3 a software component resident on the media and configured to:
4 establish trust with a media playback application;
5 establish a secure channel with the media playback application by
6 providing a public key associated with the software component to the
7 media playback application and receiving back, from the media playback
8 application, encrypted data that has been encrypted with the public key, the
9 encrypted data comprising:
10 a random number previously provided by the software
11 component;
12 a data integrity key; and
13 one or more starting numbers;
14 use the secure channel to receive a command message from the
15 media playback application, the command message comprising a data
16 section that contains a command, and an authentication section that
17 contains data that can be used to authenticate the command;
18 use the secure channel to receive status requests from the media
19 playback application; and
20 use the secure channel to send a status message to the media
21 playback application, the status message comprising a data section that
22 contains status information, and an authentication section that contains data
23 that can be used to authenticate the status information.
24
25

1 **27.** The system of claim 26, wherein said one or more starting numbers
2 comprise a starting status sequence number and a starting command sequence
3 number, said numbers being useable to ascertain, respectively, whether a status
4 message or a command message has been lost.

5
6 **28.** The system of claim 26, wherein the authentication sections of the
7 command message and the status message comprise data that has been processed
8 using the data integrity key.

9
10 **29.** The system of claim 26, wherein the command message contains a
11 command instructing the software component to enable one or more of a number
12 of different types of content protection technologies to protect media content that
13 is provided over a physical connector.

14
15 **30.** The system of claim 26, wherein the command message contains a
16 command instructing the software component to enable one or more of a number
17 of different types of content protection technologies to protect media content that
18 is provided over a physical connector, and wherein the software component is
19 configured to enable a plurality of different types of content protection
20 technologies.

1 **31.** A computing system embodying the system of claim 26.

2
3 **32.** An application program interface (API) embodied on a computer-readable
4 media, the API comprising:

5 a first method that is callable by a media playback application for
6 establishing trust between the media playback application and a software driver
7 component;

8 a second method callable by the media playback application for setting up a
9 session key between the media playback application and the software driver
10 component;

11 a third method that is callable by the media playback application to instruct
12 the software driver component to enable one or more of a number of different
13 types of content protection technologies to protect media content that is provided
14 over a physical connector; and

15 a fourth method that is callable by the media playback application to
16 request status information from the software driver component.

17
18 **33.** The API of claim 32, wherein the first method receives back a random
19 number generated by the software driver and a digital certificate.

20
21 **34.** The API of claim 32, wherein the second method provides an encrypted
22 concatenation of a random number provided by graphics hardware, one or more
23 session keys, a starting status sequence number, a starting command sequence
24 number.

1 **35.** The API of claim 32, wherein the API is exposed by a video rendering
2 component.

3
4 **36.** A method comprising:

5 calling a device driver to create an instance of a content protection device,
6 individual content protection devices being associated with individual video
7 sessions and serving as an endpoint for communication with a playback
8 application that can send commands and status requests to the content protection
9 devices;

10 maintaining, with the device driver, a global reference count for each type
11 and level of content protection that is applied to protect content;

12 maintaining, with at least one content protection device, a local reference
13 count for each type and level of content protection applied through the content
14 protection device; and

15 adjusting the global and local reference counts in accordance with changing
16 content protection types or levels.

17
18 **37.** A software architecture comprising:

19 one or more computer-readable media;

20 software driver code embodied on the computer-readable media and
21 configured to implement multiple content protection devices that are associated
22 with individual video sessions and which serve as an endpoint for communication
23 with a playback application that can send commands and status requests to the
24 content protection devices, wherein the software driver code comprises:

1 a first method that can be called to determine if a driver supports
2 content protection devices for a given output connector;

3 a second method that can be called to create an associated content
4 protection device; and

5 a third method that can be called to determine a length associated
6 with a graphics hardware certificate and to start a video session;

7 wherein individual content protection devices support callable methods
8 comprising:

9 a first method to query a graphics hardware certificate length;

10 a second method to return a variable length graphics hardware
11 digital certificate;

12 a third method for receiving a concatenation of a data integrity
13 session key, a starting status sequence number and a starting command
14 sequence number all of which are encrypted with a public key associated
15 with the graphics hardware;

16 a fourth method for receiving a command to change content
17 protection on a physical connector associated with the content protection
18 device; and

19 a fifth method for querying information about the physical connector
20 being used, the type of protection that can be applied to content being
21 transmitted through the physical connector, and the current protection level
22 that is active on the physical connector.

1 38. The architecture of claim 37, wherein the content protection device's first
2 method maps directly to the software driver code's third method.

3
4 39. The architecture of claim 37, wherein the content protection device's
5 second method maps directly to the software driver code's third method.

6
7 40. The architecture of claim 37, wherein the content protection device's third
8 method maps directly to the software driver code's third method.

9
10 41. The architecture of claim 37, wherein the content protection device's fourth
11 method maps directly to the software driver code's third method.

12
13 42. The architecture of claim 37, wherein the content protection device's fifth
14 method maps directly to the software driver code's third method.

15
16 43. A system comprising:

17 means for establishing a secure communication channel between a media
18 playback application and a component downstream from the media playback
19 application; and

20 means for using the secure communication channel to at least enable the
21 media playback application to instruct the downstream component to enable one
22 or more of a number of different types of content protection technologies to
23 protect media content that is provided over a physical connector.

1 **44.** The system of claim 43, wherein said downstream component comprises a
2 software component.

3
4 **45.** The system of claim 43, wherein said downstream component comprises a
5 hardware component.

6
7 **46.** The system of claim 43, wherein said downstream component comprises a
8 graphics hardware component.

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25